

New Direction In CAD Tools

S. Peterson
Mentor Graphics Corporation
5295 South 300 West
Murray, Utah 84065

Abstract-

Chip and software development communities are concerned with the run-away pace of chip development and the monumental problems which can quickly accumulate and overwhelm those charged with the task of IC development. This "weight" must be transferred from the shoulders of the human developers to the software tools which assimilate and control the defining data of the development process. This paper details many of the problems which are being resolved by reviewing the obstacles and resultant advancements which the industry has made. Specifically, it will describe many of the things which are being done in one company to provide this much-needed relief.

1 Introduction

Our industry's ability to develop complex ICs has grown at colossal rates as witnessed by many of the large devices currently in production. This ability to pack functionality on a chip continues to increase, limited only by the imaginations of dreamers. It's promising to note that researchers are optimistic that there are still plenty of advances to be made which will propel this industry toward new heights. Similarly, there are indications from marketing groups that as long as the technology keeps coming mankind will be able to apply it.

Obstacles however, are combining to make the task of engineering like products difficult at best with many of the development tools available today. We will address some of these in detail.

Many software tool developers are stepping up to the challenges by becoming involved in the development process. This is often accomplished through structured alliances between chip development houses and software tool developers and has, of course, been happening all along. But those building the tools have realized that the industry has moved so rapidly that unless there is a great deal of interaction with the engineers on the front lines it will be impossible to create the next generation of wonder chips. Therefore, new partnerships are being formed now and will continue to be needed to respond to the needs.

2 Outlining Some Deficiencies

There are several factors which substantially impede a development team's talent. Bear in mind that development of large chips is no longer the domain of one or two people.

Typical groups today are staffed by dozens of individuals. A team effort is needed to engineer today's system-level chips. Coordination of all this talent falls on the shoulders of a core of managers, each overseeing specific pools of responsibility. This partitioning continues on up the management tree. The hierarchy of project management begins to resemble the structure of the chip or the object of the design itself.

Note that the design process borrows heavily from the discipline of management. Accessing, controlling, and direction human engineering resources will always be an arduous task. But, the application to recorded engineering information should not be nearly that difficult, and yet in many engineering departments it remains a major stumbling block. Management of design data itself, then, is an item to be addressed.

There are other areas which have complicated chip design and layout as well. the following categories merit special attention due to the unique problems which each represents.

2.1 Process Evolution

Research and Development has been benevolent to those with aspirations of ultra-complex chips. Among others, improved lithography [1] and Statistical Process control have pushed the processing edge to where electrical lengths are less than 0.5 microns, and interconnect pitch with multiple levels of metal has crossed 2 microns [1]. Vertical stacking of devices will also soon be available [2].

If all that isn't enough (and it's not) processes have matured to the point where 15 to 25 reticule layers are capable of yielding a twin-will CMOS process which sports bipolar, electrically-erasable, polysilicon fuse, and high-voltage capabilities. Clean rooms have become so clean that there have been instances where all chips from a given wafer were fully functional.

Such amazing advances will of course continue and become more prevalent.

2.2 Complexity

Only ten years ago, a chip in a 40-pin DIP with five or six thousand transistors was considered elegant. The only thing more spectacular was the spontaneous celebration observed when the chip came out of fab and worked the first time. It happened, but not as often as hoped.

At this writing, there are many organizations with operational chips containing one to two million transistors and operation above 100 Mhz. There are devices currently being developed with more than four million devices on board, with that figure continuing to mount. These are digital "systems" chips. All you add is power (very little of it) and some I/O. Other groups have been successful in making complementary analog and digital systems, such as high-speed modems, coexist harmoniously on the same die.

This industry has even had to develop its own packaging materials and methods to answer the advancing needs of interconnect which a device needs to talk to its neighbors.

2.3 Resolution

The dimensions at which circuits are laid out have begun to outstrip needed computer resources. Shape boundaries are still being stored using integer mathematics, but as geometries begin to take on all-angle attributes to squeeze circuitry into ever small areas, floating point operations will increase, as will the working storage required to accommodate necessary algorithms or methods. Verification operations are already floating point intensive.

Shrinking processes started to dictate special rule sets several years ago. Two significant example issues are optical reflections during exposure, and mechanical stress within each of the physical materials deposited or grown during the process. Stress has become a significant reliability issue, and new design rules have been "invented" which cannot be accommodated in many of the current verification tools. They've simply run out of gas.

2.4 Performance Prediction

At smaller geometries, the ability to predict or simulate the performance of a small sub circuit, let alone the whole chip, has become very difficult. Indeed, it is the opinion of some that there is not enough time available to perform a thorough simulation. The engineer sometimes takes the risk that his circuit will be 'fast enough', and pushes the design on through reticle generation and fabrication to verify his hypothesis. That's an expensive test. If he has a lot of experience, sometimes he is lucky. Most of the time he is not. Keeping track of all critical paths in one's head is just impossible.

Because performance of datapath chips is often measured in MIPS, it becomes clear that the engineer needs a tool which will allow him to model sub-micron devices and associated parasitics accurately, so that he can optimize paths. This will allow him to squeeze performance from circuitry which has often been referred to as "design margin".

2.5 Time to Market

It seems that there are many industries which tend to drive themselves. IC development is one of those where the time to develop and get a device to manufacturing must be minimized. Redesigns and delays due to tools, once commonplace and accepted, now have no place in the process. If a chip cannot be delivered on time, the project's end product will usually suffer a great deal of lost market share. That goes to the bottom line as lost revenue.

3 Providing Solutions

Having set the stage with many of the problems which need to be solved, it is now possible to outline some of the possible remedies.

There are many vendors attempting to provide solutions to the problems mentioned herein. Speaking for these other developers would not be possible, being unfamiliar with

some of their key philosophies. A reasonable approach would be to discuss the issues as they are being addressed within Mentor Graphics Corporation. This will provide insight into treatment of the industry's needs from the tool developer's perspective - specifically that of the staff of Mentor Graphics.

3.1 Frameworks

A little over three years ago, Mentor Graphics began to seriously consider frameworks. Many in the industry felt that chip complexity was beginning to level off. Several discussions with key customers, however, drove home the point that soon the complexities of planned chips would exhaust the capabilities of current tools. Mentor Graphics not only realized this, but also faced early the grim realization that some of its planned products and future enhancements would not be able to meet many of these needs.

This was when Mentor Graphics began to develop the idea that a heavily-supported environmental approach might fill this need. A decision was made that a "from the ground up" approach was the only realistic one, and that a new language would be required to meet these needs. In this environment, individual tools could operate autonomously, yet communicate effectively with other tools at levels of abstraction defined by the user. Today, there is a great deal of activity in defining frameworks, but Mentor Graphics has made significant headway not only in defining the system, but also in establishing the standards and methods to make it work. And it's about to be released. The result is a framework called the **Falcon FrameworkTM** for concurrent design. Some of the highlights follow.

MotifTM is the graphical user interface standard developed by the Open Software Foundation (OSF) which brings a common "look and feel" to all tools intended to operate within the Falcon Framework. The compliance standards [3] conform to the **OSF/MotifTM Style Guide** (Current revision). This approach eliminates much of the need to relearn when going from tool to tool, and when new tools are added to the framework.

History has shown that those industries which provide their goods and services with an "open" attitude are usually the ones which take root and grow. This has been demonstrated many times in the hardware and software areas with which most will be familiar. It seems obvious that if any framework is going to be successful, one must realize that openness is paramount. There will be big players, and there will be small niche players. To that end, Mentor Graphics has established an **Open DoorTM** policy providing any software tool developer or customer all he or she needs to be successful in integrating his tool into the framework.

A **Design Management EnvironmentTM** allows users to control the design as well as its management process through an object-oriented paradigm. Tools and data are created as objects in which version control can be applied to both. The historical benefits as applied to design chronology are indispensable.

The **Falcon FrameworkTM** allows the user community to develop tools to assimilate objects or groups of objects in one environment. The decision support paradigm allows the engineer to rapidly determine how a design decision will affect the parameters of another part of the design. For example, thermal aspects of a chip need to be considered within

the chip, as well as in the board where it lives. While not easy to do in the past, this provides the capability of transcending such boundaries.

The **Bold™** Browser provides an integrated, on-line documentation delivery system. All 65,000 pages of Mentor Graphics' documentation are provided. The user can also create his own documentation, and include it with the system. Linked Hypertext is part of its capability.

3.2 Object-Oriented Methods

Already mentioned, the decision was made to embrace a language which would facilitate the framework thrust. The decision to go with C++ was almost automatic. An object-oriented approach was needed to support Falcon's virtues, and yet performance was paramount. Project for example, the fact that multi-million-transistor chips might involve more than a billion polygons or shapes, not to mention the volume of engineering data collected to form specifications, netlists, simulations, and the like.

There are quite a few tool developers who have felt that going to C++ represents too significant an investment: that C++ is not worth the re-education process which would be required. Most of them are still developing products in the same, standard languages today. There is no doubt that the cost of embracing C++ is substantial. Unofficially, Mentor Graphics has spent more than seventy-five million dollars on Falcon, much of that has gone into the effective adaptation of the object-oriented capabilities of C++. But this learning process has paid off in the staff's ability to write reusable code. A discussion of some of the capabilities and features of C++, as applied to design, is in order.

Variables defined in most modern languages are typed. In procedural languages, such as C, it is the programmer's responsibility to assure that improper procedures are not applied to variables. Misapplication, at the very least, results in incorrect data but often causes system crashes, and debugging is difficult.

C++ allows one to build a class, which when instantiated represents an object. A `logic_elem` (logical element) might represent a class of objects, for example, while `comb_elem` (combinational element) and `mem_elem` (memory element) would both be subclasses of `logic_elem`. NOR and NAND gates are elements which would belong to the `comb_elem` class. An entire hierarchy of gates could be developed, but that need not be done here. Other references [4,5,6,7] can provide the needed details to interested parties.

A key attribute of C++ is that rather than applying a procedure to an object of concern, operative messages are sent to the object to perform functions. Through its classification, the object has been given the intelligence which it will need to interpret or understand the message which has been sent to it. This capability exists because the object always carries its class with it, the same way that a human being carries its own gender information. To do this, it is necessary that an object's class contain not only the instantiated information which may make it unique, but also contains the procedures which will be used to manipulate its data. All of this information is encapsulated as part of the class.

The encapsulated information is broken down into categories of accessibility. One might define functions which are private, as well as functions which are public. An example of a

public operation on a logical NOR gate would be to place() a gate at a specific location in the physical domain. A private function might be to checkPlacement() to assure that the provided location is allowable. Again, these functions are a defined part of the class of an object.

This represents the briefest of introductions. There are many other virtues to consider[9]. The point of the discussion is to show that object-oriented methods represent a significant departure from the 1980's methods of programming. But application to design data is an area where the management of objects such as cells, gates, views, and similar elements does provide the methods needed to manage such vast amounts of data which are seen even now.

3.3 Verification

Given information presented in the first part of this paper, it might be obvious that there are several capabilities from previous sections which could easily be taken advantage of and applied to the task of design verification. As seen from this perspective, the possibilities take on their own characteristics. Some of the highlights follow.

True hierarchy will allow one to correct a cell which contains errors, rather than having to examine each instantiation of that cell to determine if a cited error is real or redundant. Thanks to object-oriented methods, the process of error reporting will become much more meaningful. Since redundancies will be eliminated, True errors will stand out.

Significantly enhanced polygonal operations are being implemented. Within weeks multiple tools will be available which will provide this extended checking capability.

The Design Management EnvironmentTM operating in the Falcon Framework environment mentioned earlier promises to facilitate a great deal of incremental checking which has not been commercially available until now.

3.4 Parasitic Extraction

Chip complexity has begun to severely impact many aspects of chip performance. Diminishing geometries have contributed toward positive performance in areas such as lower l-effectives, gate-oxide thicknesses, power supply voltages, and similar areas. The converse of this, however, is that reduced geometries cause other problems with increased fringe capacitance and interconnect resistance.

Mentor Graphics will soon ship a tool with significant parasitic capacitance and resistance extraction capability. This tool will fill the need of accurate extraction and consideration of parasitics during simulation. Per existing customer communications, it is expected to be of substantial value when designing cells or libraries of cells. It will also facilitate the analysis of global signal nets, such as a clock and asynchronous reset lines. Capacitance extraction is derived from the boolean operations which are already available within the checking environment previously discussed. Resistance calculations are performed based upon optimized heuristic methods to enhance throughput.

Further research continues in an attempt to satisfy the needs of analog and bipolar designers. High resistance and capacitance extraction accuracy, while not critical in digital applications, is an absolute must in the mixed-signal world. Finite element methods, where the customer feels that he needs the accuracy, are expected to provide the needed accuracy of resistance extraction.

Similarly, the high-speed digital developers will soon find themselves in need of tools capable of extracting segmented transmission lines to analyze crosstalk and square up interconnect carrying clock signals. These will certainly have to be provided as well.

4 Conclusions

The information herein has given a representative snapshot of current efforts of IC design and development automation, as seen by those currently involved in the development of these tools. While the software industry is stepping up these challenges, it is aware and concerned with the obvious: keeping up with chip complexity has, to date, only been a hope. The goal is to somehow get ahead of the currently accelerating capabilities. While the few solutions presented herein will hopefully meet near-term needs, recent history has demonstrated that the only way that this can be realized is through significant interaction between the IC design and tool development communities.

References

- [1] J.H. Brannon, "Eximer Laser Ablation and Etching", IEEE Circuits and Devices, 6, #5, pp. 19-24, 1990.
- [2] G.W. Neudeck, "Three-Dimensional CMOS Integration", IEEE Circuits and Devices, 6, #5, pp. 32-38, 1990.
- [3] Mentor Graphics UIMS/Motif Compliance Checklist,,, Revision 1.0.1, 23 August 1990, Mentor Graphics corporation.
- [4] B. Stroustrup, "What is "Object-Oriented Programming?"", Selected Readings, 307-144, pp. 4-1-4-25, 1989.
- [5] B. Stroustrup, "The C++ Programming Language", Addison- Wesley, 1987.
- [6] S.B. Lippman, "C++ Primer", Addison-Wesley, 1989.
- [7] T. Hansen, "The C++ Answer Book", Addison-Wesley, 1990.
- [8] D. Thomas, "What's in an Object?", Byte, 14, #3, pp.231- 240, 1989.